
DATAMAC BASIC (TINIMAC)

The logo features the word "data" in a bold, italicized sans-serif font, followed by a stylized bracket-like symbol that connects to the word "mac" in a similar bold, italicized sans-serif font.

data* *mac

USER MANUAL

**DATAMAC BASIC
(TINIMAC)
USER MANUAL**

**DATAMAC COMPUTER SYSTEMS
680 ALMANOR AVENUE
SUNNYVALE, CA. 94086**

TABLE OF CONTENTS

SECTION

1.0	INTRODUCTION.....	1
2.0	GENERAL INFORMATION.....	2
3.0	LOADING TINIMAC.....	3
4.0	USING THE EXPLAIN FEATURE.....	4
5.0	TINIMAC BASIC CATEGORIES.....	5
5.1	Program Statement.....	5
5.2	Arithmetic Functions.....	7
5.3	String Functions.....	7
5.4	Commands.....	8
5.5	Special Functions.....	9
5.6	Input/Output Functions.....	10
6.0	EDIT MODE.....	10

DATAMAC BASIC

(TINIMAC)

USER GUIDE

1.0 INTRODUCTION

TINIMAC basic includes all necessary tools for basic programming and some unique features that apply only to the DATAMAC computer. The user should also be aware that TINIMAC can only be used in conjunction with the DATAMAC computer.

A special feature of TINIMAC, when used in conjunction with a DATAMAC 800 series computer, is its ability to perform remote diagnostics via modem connection for program debugging. TINIMAC also has its own self-contained mini-book (EXPLAIN Function)(Ref. Section 4.0) that is an explanation of all basic statements and commands contained on the disk.

NOTE:

TINIMAC basic does not have the capability to handle transcendental functions (COS,SIN,LOG etc.) or floating point.

2.0 GENERAL INFORMATION

NUMBERS

All numbers are integers and must be between -32767 and 32767.

VARIABLES

There are 26 variables denoted by letters A through Z. There are also 26 arrays A(I). The dimension of this array (i.e. The range of values of the index I) is set by the DIM verb and is only one (1) -- not two (2) dimensional.

FUNCTIONS

A full list of functions and commands may be listed by using the VLIST verb and a detailed explanation by using the EXPLAIN verb per the following format: EXPLAIN VERBNAME.

EXPRESSIONS

Expressions are formed with numbers, variables and functions with arithmetic and compare operators between them. + and - signs can also be used at the beginning of an expression. The value of an expression is evaluated from left to right, except that * and / are always executed first, + and - second and compare operators last. Parentheses can also be used to alter the order of evaluation.

STATEMENTS

A TINIMAC statement must consist of a statement number between the numbers 1 and 32767 followed by one or more commands. Commands in the same statement are separated by a semi-colon. ";", GOTO, STOP and RETURN commands must be the last command in any given statement.

PROGRAM

A TINIMAC program consist of one or more statements. When a direct command RUN is issued, the statement with the lowest statement number is executed first, then the one with the next lowest statement number, etc.. However, the GOTO, GOSUB, STOP and RETURN commands can alter this normal sequence. Within the statement, execution of the commands is from left to right. The IF command can cause the execution of all commands to its right in the same statement to be skipped.

COMMANDS

TINIMAC has three (3) kinds of commands within its program and each has its own particular way of being executed.

- Statement Command: A statement command is used within a program statement and can be assigned a number, put into a string and saved on disk.
- Direct Statement: A direct command is one that is executed from the keyboard in which immediate action is taken and that action is displayed onto the display screen. This kind of command cannot be given a number; cannot be put into a string with other commands; and it is re-executable.
- Diagnostic Command:
For DATAMAC internal use only.

* CAUTION *

THIS KIND OF COMMAND WILL READ AND WRITE ONTO
YOUR DISK - USE AT YOUR OWN RISK. DATAMAC WILL
NOT ASSUME ANY RESPONSIBILITY FOR DESTROYED SOFT-
WARE DUE TO USER USE OF THESE COMMANDS

The diagnostic commands are special purpose commands used to directly control the DATAMAC 800 series computer hardware and to have direct control of the 5 1/4 inch floppy disk environment. These commands are also used in conjunction with a modem to perform remote diagnostics and program debugging for the user.

The following is a list of Diagnostic Commands appearing on the VLIST and in the EXPLAIN section of TINIMAC.

DRIVE	HOME	SEEK	HEAD
DSKBUF	ENTER	JUMP	DUMP
WRSECTOR	RDSECTOR	BDOS	RESULT
SFCB	DMA	CON	DEFINE

COMMAND ABBREVIATION

All commands listed in the VLIST can be abbreviated down to two (2) character abbreviations followed by a period. If more than one of the commands have the same two character abbreviation, one of them must have a third character added followed by a period.

3.0 LOADING TINIMAC

PROCEDURE:

- Load the CP/M operating system.
- When the **A>** appears, type in **TINIMAC** -- depress **RETURN**.

NOTE:

ALL ENTRIES FROM THE KEYBOARD MUST BE DONE IN UPPER CASE

The following message will appear on the screen:

```
Datamac Computer Systems
--- T i n i M a c ---

T-Mac
>
```

4.0 USING THE EXPLAIN FEATURE

The EXPLAIN feature allows the user to display a complete explanation of each basic command and statement directly onto the display screen. This function can be used anytime TINIMAC is running or testing a program and does not disturb the operation.

PROCEDURE:

- Ensure that TINIMAC basic is loaded into the system.
- Type in **VLIST** -- depress **RETURN**.

The following message will appear on the screen:

```
EXPLAIN  LIST    VLIST    LLIST    BYE      EDIT      RUN
MERGE    NEW     DIR      DELETE  LOAD      SAVE      RENUM
NEXT     DIM     DUMP    FILL    ENTER    JUMP      PAUSE
BELL     CLS     VMODE   DEFINE  DMA       DRIVE     HOME
HEAD     SEEK    RESULT  WRSECTOR RDSECTOR  BDOS      SFCB
CON      CHAIN   IF      GOTO    GOSUB    ON        PRINT
LPRINT   DIM      STOP    RETURN  INPUT    LINE      LET
FOR      OUT     POKE    REM     RND      INP       FREE
PEEK     CALL    MOD     LEN     VAL      TO        STEP
THEN     @        TAB     SPACE$  INPUT$   STR$      CHR$
RIGHT$   LEFT$    MID$    STRING$ INKEY$   DSKBUF    PROGEND
>=       <>      >       =       <=      <        AND
OR
T-Mac
>
```

- Type in **EXPLAIN VERB NAME** (OUT for example) -- depress **RETURN**.

The following message will appear on the screen:

```
"OUT" SENDS DATA DIRECTLY TO AN OUTPUT PORT. THE FORMAT IS 'OUT
P,D' WHERE P IS THE PORT NUMBER AND D IS THE DATA BYTE.
```

```
T-Mac
>
```

5.0 TINIMAC BASIC CATEGORIES

TINIMAC is divided into eight(8) different categories with each statement, function or command having its own unique method of operation. The following text will be a description and listing of each category.

5.1 Program Statement

BELL -- Activates the audible tone in the CRT. The format is "BELL(N,D)" where N is the number of tones to be sounded and D is the duration of the tone and the space between tones.

CHAIN -- Is used at the end of one basic program to automatically load and run another basic program. The format is "CHAIN X" where X is the name of the next program to load.

CLS -- Clears the CRT screen and places the cursor in the upper left hand corner of the screen.

DIM -- Is used to establish the size of an array. The format is "DIM A(X)" where A is the name of the array and X is the maximum number of array elements minus 1 (all arrays begin with zero). The maximum array size must be less than half the number of bytes remaining in memory.

FOR -- Begins a "FOR...NEXT" loop which allows repetition of a series of program operations. The format is "FOR X = A TO B STEP S" where X becomes the current loop index, A is the beginning value of the index, B is the limit of the loop index and S is the value by which the loop is incremented. If "STEP S" is omitted, then the default value of the increment is +1.

GOSUB -- Forces the basic program to begin executing a subroutine. The format is "GOSUB X" where X is the number of the first line of the subroutine.

GOTO -- Sends the program to another line to continue operation. The format is "GOTO X" where X is the line number that will be jumped to.

IF -- Permits conditional branching in order to transfer program control if some condition is fulfilled. The format is "IF A THEN B" where A is the condition and B is the branch taken or value assigned if A is true.

LET -- Is used to assign a value to a variable. The format is "LET X = V" where X is the variable and V is the value. LET is optional and the statement "X = V" will do exactly the same thing.

NEW -- Erases the basic program currently loaded in memory. This command does not affect files saved to the disk.

NEXT -- Ends a "FOR...NEXT" loop. It increments the loop counter and compares it to the loop limit exiting the loop when the values are equal.

ON -- Is a conditional used with "GOTO" or "GOSUB" to transfer control of the program to a different line when the condition is true. The format is "ON X GOTO Y" or "ON X GOSUB Y" where X is the condition to be true and Y is the line number of the program where control is to be transferred.

OUT -- Sends data directly to an output port. The format is "OUT P,D" where P is the port number and D is the data byte.

PAUSE -- Provides for the delays. The format is "PAUSE X" where X is the length of delay in approximately milliseconds (i.e. "PAUSE 1000" would delay for about one second).

POKE -- Allows a value to be placed into a memory location. The format is "POKE M,V" where M is the address in memory and V is the data byte.

REM -- Is short for REMARK. Anything on the line following REM is not executed. This command is to provide a method for commenting a program.

RETURN -- Must always be used at the end of a subroutine in order to transfer control back to the main program after the subroutine is executed. (SEE "GOSUB")

RND -- Is used to generate a positive random integer. The format is "X = RND(V)" where X is the value of the number generated and N is one greater than the largest number that could possibly be generated (i.e. If N was equal to 9, the random numbers that were generated would be from 0 to 8). N must be a positive integer from 2 through 32767 (if N was 1 only, 0 would be returned).

SPACE\$ -- Positions the cursor. The format is "PRINT SPACE\$(X);Z" where X is the number of spaces to move the cursor and Z is what is to be printed. X must be an integer from 1 to 255.

STOP -- Is used as the last command in a program in order to HALT.

THEN -- Specifies the action in a conditional branch when the IF statement is true. (SEE "IF")

TO -- Is used within a "FOR...NEXT" statement between the beginning value of the loop index and the loop limit. (SEE "FOR" FOR THE FORMAT AND FURTHER EXPLANATION)

@ -- (AT) Positions the cursor within a print statement. The format is "PRINT @ (X,Y);Z" where X is the row number, Y is the column number and Z is what is to be printed at that location. Another option allows the format "PRINT @ (X);Z" where X is an

integer from 1 to 1920.

5.2 Arithmetic Functions

AND -- Performs a logical "AND" comparison in bitwise fashion upon any two values from 0 through 32767. The format is "R=A AND B" where A and B are the two values to be compared and R is the result.

MOD -- Returns the remainder after one value is divided by another. The format is "X=A/B:Y=MOD" where A is the dividend, B is the divisor, X becomes the quotient, and Y becomes the remainder.

OR -- Performs a logical "OR" comparison in bitwise fashion upon any two values from 0 through 32767. The format is "R= A OR B" where A and B are the values to be compared and R is the result.

XOR -- Performs a logical "EXCLUSIVE OR" comparison in bitwise fashion upon any two values from 0 through 32767. The format is "R = A XOR B" where A and B are the values to be compared and R is the result.

> -- Is a relational operator meaning greater than.

= -- Is a relational operator meaning equal to.

< -- Is a relational operator meaning less than.

<= -- is a relational operator meaning less than or equal to.

>= -- Is a relational operator meaning greater than or equal to.

<> -- Is a relational operator meaning not equal to.

/ -- Divide (Note: Since DATAMAC uses integers only, 2/3=0)

* -- Multiply

- -- Subtract.

+ -- Add

5.3 String Functions

INKEY\$ -- Provides a method of interrupting a program in operation by pressing a key on the keyboard. If the specified key is pressed, a flag is set so that when the INKEY\$ statement is reached in a program, that operation may be halted or control may be transferred. The format is "X\$=INKEY\$" where X\$ becomes the value of the key that is pressed. The INKEY\$ character is not echoed on the CRT.

INPUT\$ -- Allows up to 75 characters to be input from the keyboard without echo onto the CRT. The format is "X\$=INPUT\$(N)" where N is the number of characters to be input from the keyboard and X\$ becomes the characters that were input. Upon execution, program operation halts until N characters are input. No prompt is provided when the command is executed.

LEFT\$ -- Is used to return the leftmost characters contained in a string. The format is "LEFT\$(X\$,N)" where X\$ is the string and N is the number of characters to be returned counting from the left of X\$. If N is greater than the number of characters in X\$, then X\$ will be returned.

LEN -- Returns the number of characters in a string. The format is "LEN(X\$)" where X\$ is the string to be measured.

LINE -- Used with "INPUT" provides for a string input from the operator to be executed on the same line as the prompt. The format is "LINE INPUT "X\$" A\$" where X\$ is the prompt and A\$ becomes the response.

MID\$ -- Is used to return the middle characters in a string. The format is "MID\$(X\$,N,Y)" where X\$ is the string. The Nth is the first character to return and Y is the number of characters to return to the right of N.

RIGHT\$ -- Is used to return the rightmost characters contained in a string. The format is "RIGHT\$(X\$,N)" where X\$ is the string and N is the number of characters to be returned counting from the right of X\$. If N is greater than the number of characters in X\$, then X\$ will be returned.

STR\$ -- Returns the string representation of the value. The format is "STR\$(X)" where X is the value.

STRING\$ -- Returns a string of ASCII code. The format is "STRING\$(N,X)" where X is the decimal value of the ASCII code and N is the number of times it is to be returned. The maximum value for N is 127.

VAL -- Returns a value for a string of numerals so that mathematical or logical operations may be performed on them. The format is "VAL(X\$)" where X\$ is the string of numerals.

& -- Performs decimal to hexadecimal and hexadecimal to decimal conversions. Print "&1024" returns "0400". "PRINT (&400)" returns "1024". The range of conversions is from 0 through 32767 in decimal or 0 through 7FFF in hexadecimal.

5.4 Commands

BYE -- Exits basic and returns to CP/M. All programs not saved to the disk previous to this command will be lost.

DIR -- List the CP/M directory.

FREE -- Returns the number of bytes available program space remaining in working memory. The format is "PRINT FREE".

LIST -- Outputs a listing of the current basic program to the CRT screen. Press the space bar to start and stop the display. Press CONTROL C to return to the basic command level.

LLIST -- List the current basic program to the printer and the CRT screen.

MERGE -- Is used to add a program from the disk to one in memory. The format is "MERGE X" where one program is in memory and X is the name of the program to be merged from the disk. The first line number of the program to be merged must be greater than the last line number of the program in memory. All variable values are lost when a merge command is performed. Basic returns to the command level after the merge is completed. A program will not continue running if merge was a program statement.

PROGEND -- Returns the address of the program pointer (the next available address in memory). The format is "X=PROGEND" where X becomes the value of the address.

RENUM -- Renumbers the lines of the currently loaded program. The format is "RENUM X" where X is the new first line number and the new increment between lines. Any line numbers within program statements (such as after a GOTO or GOSUB command) must be updated manually with the EDIT function.

RUN -- Causes execution of the basic program currently loaded into working memory.

VLIST -- Is short for vocabulary list. This command lists all available basic commands on the CRT screen.

5.5 Special Functions

PEEK -- Allows a variable to be given the value of the content of a memory location. The format is "X=PEEK(M)" where M is the address of the memory location and X becomes the value of its content.

INP -- Is used to read a data byte directly from an input port. The format is "X=INP(P)" where X becomes the value of the data byte and P is the port number.

5.6 Input/Output Functions

DELETE -- Erases basic programs that have been saved to the disk. The format is "DELETE X" where X is the name of the file to be erased.

INPUT -- Is used to prompt the operator for information. The format is "INPUT X" where X is numeric data or "INPUT X\$" where the data is a character string.

LOAD -- Loads a basic program from the disk into working memory.

LPRINT -- Is used to have the computer type out information to the printer and the CRT screen.

PRINT -- Is used to have the computer type out information to the CRT screen. The format is "PRINT X" where X is a value, a variable representing a value, or a string variable (i.e. words). String variables must be within quotation marks.

SAVE -- Saves the current basic program in working memory to the disk. The format is "SAVE X" where X is the name the program will be saved under on the disk.

TAB -- Moves the cursor across the line on the CRT screen. The format is "TAB(X)" where X is the number of spaces to be moved.

6.0 EDIT MODE

The TINIMAC EDIT MODE allows the programmer to change characters, lines or partial lines, within a program, without having to re-type the complete line.

To bring up the EDIT function, type in the word EDIT followed by the line number to be changed, then depress RETURN.

The following is a list of the EDIT functions and a description of each:

SP (SPACE BAR) -- The space bar (SP) allows the user to either display one character at a time, of a given line, or multiple characters.

To display one character - depress the space bar once. To display more than one character, type in the number of characters to be displayed (i.e. for 6 character display, enter the number 6) and depress the space bar once.

CR (RETURN KEY) -- Depressing this key, while in the EDIT MODE, indicates to the computer that you have completed your editing of that current line. Upon depression of the RETURN key, the computer will return back to the COMMAND MODE and record your changes.

A -- Abort present line editing and restart at the beginning of the same line without making the previous corrections.

C -- Change one character. When the C key is depressed, the computer will allow you to change the character at the current cursor position only. If you wish to change more than one character, type in the desired number of characters then depress the C key.

D -- Delete the next character. This function allows you to delete the next character directly to the right of the cursor. To delete more than one character, type in the number to be deleted (to the right of the cursor), then depress the C key.

E -- Exit EDIT MODE without printing the line. This function allows you to end the editing of the present line and all prior lines and to save all changes that were made. This function saves only the changes in working memory which means the SAVE function must be executed in order to save the changes onto the disk.

H -- Delete all characters to the right of the cursor position. When the H key is depressed, all characters to the right of the cursor are deleted; however, you are still in the H function and can now enter characters (to the right of the cursor) to replace the ones that were deleted.

I -- Begin insert mode at this point. This function allows you to enter characters, to the right of the cursor position, within the existing program line.

K -- Search for character and kill. To utilize this function, you must first determine if the character appears more than once in the line. When executed, this function will delete all characters up to but not including the one designated. If the character appears more than once, you would type in the following: 2Kc (c= the search character).

L -- Display line to be edited. When you tell the computer to edit a line, only the line number will appear. To display the text of the line, depress the L key. The computer will display the line text, move the cursor down to the next line and allow you to edit the line above.

Q -- Stop edit - do not update line. Cancels all changes made to the current line and exits EDIT MODE.

S -- Search for character. Unlike the K function that searches and deletes characters, the S function searches for the character(s) specified and leaves all characters intact. You also have to specify the number of character occurrences if there are more than one. The S function starts at the current cursor position in the line and starts its search to the right until the character is found or to the end of the line if the character is not found.

X -- Display line and insert character at the end. Upon depression of the X key, the computer will display the line specified, move the cursor to the end of the line and allow you to insert more characters from the point of the cursor.

ESC -- End insert mode. This function turns off the insert function I and allows you to continue with other functions.

